

UNITED STATES PATENT APPLICATION

FOR

PROTECTING ACCESS TO MICROCONTROLLER MEMORY BLOCKS

Inventors:

Warren Snyder

Prepared by:
WAGNER, MURABITO & HAO, LLP
Two North Market Street
Third Floor
San Jose, California 95113
(408) 938-9060

1
2
3
4 **PROTECTING ACCESS TO MICROCONTROLLER MEMORY BLOCKS**
5
6
7

8 **CROSS REFERENCE TO RELATED DOCUMENTS**

9 This application is related to and claims priority benefit of U.S. Provisional
10 Patent Application Serial No. 60/243,708, filed October 26, 2000 to Snyder, et al.
11 which is hereby incorporated herein by reference.
12

13 **COPYRIGHT NOTICE**

14 A portion of the disclosure of this patent document contains material which
15 is subject to copyright protection. The copyright owner has no objection to the
16 facsimile reproduction of the patent document or the patent disclosure, as it
17 appears in the Patent and Trademark Office patent file or records, but otherwise
18 reserves all copyright rights whatsoever.
19
20

21 **FIELD OF THE INVENTION**

22 This invention relates generally to the field of memory security, for example
23 in microcontrollers and the like. More particularly, this invention relates to a
24 method and apparatus for mapping memory blocks to security levels to limit access
25 thereto and for modifying such security levels.
26

27 **BACKGROUND OF THE INVENTION**

28 Microcontrollers such as 100 depicted in **FIGURE 1**, generally include a
29 processor (Central Processing Unit) 106 and associated Random Access Memory

1 (RAM) 110 as well as a block of nonvolatile memory 116, generally flash memory,
2 used to store a user program. By using a block of user programmable nonvolatile
3 memory 116, the microcontroller may be customized to carry out any desired
4 function within the capabilities of the device. Numerous techniques exist for
5 programming the user program into the nonvolatile memory 116. In general, such
6 techniques may be characterized by use of an external tester/programmer 120
7 coupled directly to the nonvolatile memory 116. The tester/programmer 120 utilizes
8 a control signal line 124 to appropriately signal the nonvolatile memory 116 (as well
9 as associated circuitry within the microcontroller 100) that a programming mode
10 is being entered. An address line, path or bus 130 is then used to identify memory
11 locations within the nonvolatile memory 116 being programmed. Data is
12 transmitted to the nonvolatile memory 116 over a data line, path or bus 138. When
13 all address locations have been appropriately programmed within the nonvolatile
14 memory 116, the tester/programmer 120 issues appropriate control signals on
15 control path 124 to terminate the programming process. Thus, the
16 tester/programmer 120 can directly manipulate the memory 116 with nothing to
17 prevent unauthorized tampering with or copying the memory content.

18 The details of the exact programming process vary from manufacturer to
19 manufacturer and from part to part. However, the above characterization generally
20 describes the process used. Unfortunately, the process of programming the
21 microcontroller 100 as depicted in **FIGURE 1** presents a number of problems. The
22 need to bring control data and address lines to the outside requires that the
23 microcontroller 100 frequently have more I/O (input/output) pads on the processor
24 than might otherwise be necessary. In addition, the external accessibility to control
25 path 124, address path 130 and data path 138 renders microcontroller 100
26 susceptible to unauthorized memory reads, programming or reprogramming. This
27 may present a serious security problem making microcontroller 100 vulnerable to
28 unauthorized modification of a user program including potentially infecting the user
29 program with "bugs" and "viruses".
30

SUMMARY OF THE INVENTION

The present invention relates generally to controlling access to memory blocks. Objects, advantages and features of the invention will become apparent to those skilled in the art upon consideration of the following detailed description of the invention.

In one embodiment of the present invention a microcontroller provides protection to memory blocks in an embedded memory. A set of rules such as security levels mapped to memory blocks are stored in a nonvolatile memory. An algorithm for application of the rules is stored, for example, in a supervisory ROM. When a read or write operation is to be carried out, the rules are applied according to the algorithm in order to authorize or reject the read or write operation. Security levels can be modified, according to defined rules. In one embodiment, the security levels can only be increased. This provides many advantages including the advantages that the memory security cannot be decreased below a default value established by the user, and intruders may be thwarted from attempts to modify the memory content in secure memory blocks.

In a microcontroller consistent with embodiments of the present invention, a method of accessing a block of memory includes, in response to an access request to the block of memory, ascertaining a security rule associated with the block of memory; applying the security rule according to a security algorithm to determine if the access request is authorized; and denying the access request in the event the access request is unauthorized.

In a microcontroller having an internal processor, a method of accessing memory according to other embodiments of the invention includes mapping a security level to each block of a plurality of memory blocks and storing the mapping in a table; in response to an access request to a specified block of memory, determining the security level for the specified block of memory; applying the a security algorithm using the security level to determine if the access request is authorized by the algorithm; and denying the access request in the event the

access request is unauthorized.

In a microcontroller, a method of controlling secure access to memory according to other embodiments of the invention include, in response to a request to change a security rule for a block of memory, determining if the change in the security rule for the block of memory is authorized; and denying the request in the event the request is unauthorized.

The above summaries are intended to illustrate exemplary embodiments of the invention, which will be best understood in conjunction with the detailed description to follow, and are not intended to limit the scope of the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the invention believed to be novel are set forth with particularity in the appended claims. The invention itself however, both as to organization and method of operation, together with objects and advantages thereof, may be best understood by reference to the following detailed description of the invention, which describes certain exemplary embodiments of the invention, taken in conjunction with the accompanying drawings in which:

FIGURE 1 is a block diagram illustrating a conventional microcontroller with external access to memory.

FIGURE 2 is a block diagram of a microcontroller consistent with an embodiment of the present invention.

FIGURE 3 is a flow chart describing a process of accessing protected memory blocks consistent with an embodiment of the invention.

FIGURE 4 is a flow chart describing another process of accessing protected memory blocks consistent with an embodiment of the invention.

FIGURE 5 is a flow chart describing a process of modifying security levels for memory blocks consistent with an embodiment of the present invention.

1 **FIGURE 6** is a flow chart describing another process of modifying security
2 levels for memory blocks consistent with an embodiment of the present invention.
3
4

5 **DETAILED DESCRIPTION OF THE INVENTION**

6 In the following detailed description of the present invention, numerous
7 specific details are set forth in order to provide a thorough understanding of the
8 present invention. However, it will be recognized by one skilled in the art that the
9 present invention may be practiced without these specific details or with
10 equivalents thereof. In other instances, well known methods, procedures,
11 components, and circuits have not been described in detail as not to unnecessarily
12 obscure aspects of the present invention.
13
14

15 **NOTATION AND NOMENCLATURE**

16 Some portions of the detailed descriptions which follow are presented in
17 terms of procedures, steps, logic blocks, processing, and other symbolic
18 representations of operations on data bits that can be performed on computer
19 memory. These descriptions and representations are the means used by those
20 skilled in the data processing arts to most effectively convey the substance of their
21 work to others skilled in the art. A procedure, computer executed step, logic block,
22 process, etc., is here, and generally, conceived to be a self-consistent sequence
23 of steps or instructions leading to a desired result. The steps are those requiring
24 physical manipulations of physical quantities.

25 Usually, though not necessarily, these quantities take the form of electrical
26 or magnetic signals capable of being stored, transferred, combined, compared, and
27 otherwise manipulated in a computer system. It has proven convenient at times,
28 principally for reasons of common usage, to refer to these signals as bits, values,
29 elements, symbols, characters, terms, numbers, or the like.

1 It should be borne in mind, however, that all of these and similar terms are
2 to be associated with the appropriate physical quantities and are merely convenient
3 labels applied to these quantities. Unless specifically stated otherwise as apparent
4 from the following discussions, it is appreciated that throughout the present
5 invention, discussions utilizing terms such as "processing" or "computing" or
6 "translating" or "calculating" or "determining" or "scrolling" or "displaying" or
7 "recognizing" or the like, refer to the action and processes of a computer system,
8 or similar electronic computing device, that manipulates and transforms data
9 represented as physical (electronic) quantities within the computer system's
10 registers and memories into other data similarly represented as physical quantities
11 within the computer system memories or registers or other such information
12 storage, transmission or display devices.

13 14 15 **PROTECTING ACCESS TO MICROCONTROLLER MEMORY BLOCKS IN** 16 **ACCORDANCE WITH THE INVENTION**

17 While this invention is susceptible of embodiment in many different forms,
18 there is shown in the drawings and will herein be described in detail specific
19 embodiments, with the understanding that the present disclosure is to be
20 considered as an example of the principles of the invention and not intended to limit
21 the invention to the specific embodiments shown and described. In the description
22 below, like reference numerals are used to describe the same, similar or
23 corresponding parts in the several views of the drawings.

24 Referring now to **FIGURE 2**, a microcontroller 200 consistent with an
25 embodiment of the present invention is illustrated in which processor 206 is
26 coupled to internal Random Access Memory (RAM) 210. Nonvolatile memory 216
27 (e.g., Flash memory) for storing a user program is connected to the processor 206
28 internally. In addition, a supervisor nonvolatile memory 218 is also connected to
29 processor 206. A control path 224 is provided from processor 206 to nonvolatile

1 memory 216 and 218. In addition, conventional address path 230 and data path
2 238 from processor 206 to memory 216 and 218 is provided. In addition to the
3 RAM 210 and nonvolatile memories 216 and 218, microcontroller 200 includes a
4 control program ROM (a supervisor ROM) 244 that contains hard-coded
5 instructions used by processor 206 to directly program memory locations of
6 nonvolatile memory 216. Thus, all programming of nonvolatile memory 216 is
7 carried out by an internally stored process executing as a program on processor
8 206.

9 A tester/programmer 220 may be coupled to microcontroller 200 via a path
10 250 which, in the preferred embodiment, is a two wire bus carrying control, address
11 and data information to an internal test/control interface 260. The test/control
12 interface 260 is coupled to processor 206 to provide the processor with program
13 instructions and other test and control function instructions from the
14 tester/programmer 220. Instructions from the tester/programmer 220 are stored in
15 an instruction queue 268 for sequential retrieval and execution by processor 206.

16 The user program is stored in nonvolatile memory 216 which is arranged in
17 memory blocks. In one embodiment, this memory 216 provides 16 Megabytes of
18 storage for user program and is divided into 256 memory blocks. Of course, this
19 particular arrangement is to be considered exemplary since any other arrangement
20 could also be used with the present invention.

21 The supervisory nonvolatile memory 218 contains a table mapping a security
22 level for each of the memory blocks in the user program nonvolatile memory 216.
23 In one embodiment, four security levels 0 through 3 are defined as illustrated in
24 **TABLE 1** below.

Security Level	Definition
0	unsecure - no read protect, no write protect
1	read protect, write protect, processor can self modify
2	read protect, no processor self modify, no write protect
3	full read protect, full write protect

TABLE 1

According to this security definition, security level 0 is the least secure and security level 3 is the most secure. It is, however, somewhat ambiguous whether security level 2 is more or less secure than security level 1, since the actual security of these middle levels is somewhat application dependent. In this embodiment, the security levels may be defined by setting two bits in a memory table. The number of bits can be adjusted if more or fewer security levels are to be used.

The supervisor nonvolatile memory 218 stores a table similar to **TABLE 2** below to define the security level of each of the 256 blocks of memory 216. The security levels presented are presented for illustrative purposes and do not represent a default security level or have any other particular significance.

Block Number	Security Level
0	3
1	2
2	1
3	1
4	0
...	...
255	2

TABLE 2

In this example, block 0 is shown to have the highest security level, while block 4 has the lowest security level. Blocks 1 and 255 have security level 2 and blocks 2 and 3 have security level 1. In accordance with embodiments of the present invention, these security levels are programmed at the time of manufacture in accord with the end user's requirements. The security levels can subsequently be changed under certain circumstances either by programming using tester/programmer 220 or through actions of processor 206 operating under program control. However, as a security measure, the security levels can only be changed in a prescribed manner.

In general, in accordance with embodiments of the present invention, the security levels assigned to the memory blocks can only be changed under control of an algorithm stored in the supervisor ROM 244 in accordance with rules stored in the supervisor nonvolatile memory 218. Such rules in memory 218, in accordance with one embodiment define the security levels and permissible transitions in security levels as will be described later. However, in general, passwords, unique identifications for users, encryption and other security measures

could equally well be applied using the general principles of the present invention.

Consistent with the exemplary embodiment shown, the security level can only be increased, not decreased. In the present example, wherein there exists ambiguity in the absolute security of intermediate security levels, changing from one intermediate level to the other is not permitted. Thus, the permissible transitions in security levels is described in **TABLE 3** below for the current example. Those skilled in the art will appreciate that a greater or lesser number of security levels and differing security definitions can be utilized without departing from the present invention.

Security Transition	Transition Allowed?
0 to 1	YES
0 to 2	YES
0 to 3	YES?
1 to 0	NO?
1 to 2	NO
1 to 3	YES
2 to 0	NO?
2 to 1	NO
2 to 3	NO?
3 to 0	NO?
3 to 1	NO?
3 to 2	NO?

TABLE 3

In one embodiment, these permissible transitions, as defined in **TABLE 3**, can be hard coded within the algorithm in supervisor ROM 244 or may form a type of rule defined by the supervisor nonvolatile memory 218 that is referenced in

1 carrying out the security measures of the present invention.

2 By way of example of an embodiment of the invention, the security level of
3 memory block 4 is initially assigned as 0. The security level could be increased
4 to level 2 by action of tester/programmer 220 and then subsequently increased to
5 security level 3 by action of the tester/programmer. Once the security level reaches
6 the highest level (3 in this example), it cannot be modified. At this point, the only
7 way to modify the security level to reduce it is to erase the supervisor nonvolatile
8 memory 218 and re-initialize the microcontroller. This action will result in the user
9 defined default security levels being restored. In this case, the security level of
10 memory block 4 would obviously return to 0 since there is no way to reach this
11 insecure level except by it being so defined as a user default.

12 Referring now to **FIGURE 3**, a process 300 of providing security to memory
13 blocks within nonvolatile memory 216 in accordance with an embodiment of the
14 present invention starts at 304. When an instruction is received at 310, it is
15 determined at 316 whether the instruction is a read or write operation to a protected
16 memory block within nonvolatile user memory 216. If not, normal processing of the
17 instruction is carried out at 320 and control returns to 310 to await the next
18 instruction. However, if the instruction is a read or write instruction to protected
19 memory 216 at 316 control passes to 324 where the memory block is identified for
20 the read or write operation.

21 At 330, the memory block is looked up in the protection table of supervisory
22 nonvolatile memory 218. The processor 206 then determines through the
23 supervisor programs stored in supervisor ROM 244 whether or not the read or write
24 operation is authorized at 338 based upon the security algorithm residing in ROM
25 244 and the rules defined in memory 218. If the read or write operation is
26 authorized at 338, then the operation is executed at 344 and control returns to 310.
27 However, in the event the operation is not authorized at 338, the operation is
28 rejected and an error condition exists at 352. Any number of actions can take place
29 as a result of such an error condition at 352 including shutdown of the
30 microcontroller 200, presenting an error message, presenting an alert or any other

1 suitable action. In certain embodiments, control may then be passed from 352
2 back to 310 to await the next instruction.

3 Process 300 of **FIGURE 3** can be generalized as process 400 of **FIGURE**
4 **4**. When an instruction is received at 310, it is determined at 316 whether the
5 instruction is a read or write operation to a protected memory block within
6 nonvolatile user memory 216. If not, normal processing of the instruction is carried
7 out at 320 and control returns to 310 to await the next instruction. However, if the
8 instruction is a read or write instruction to protected memory 216 at 316 control
9 passes to 324 where the memory block for the read or write operation is identified.
10 At 410, security rules for the identified memory block are read from the supervisory
11 nonvolatile memory 218. These rules are applied to determine if the read or write
12 operation is permitted using the security algorithm stored in supervisory ROM 244
13 at 420. If, upon applying the rules using the security algorithm, the read or write
14 operation is permitted at 430, then the read or write operation is executed at 344
15 and control returns to 310 to await the next instruction. However, if the read or write
16 operation is not permitted at 430, the operation is rejected and an error condition
17 exists at 352.

18 As described previously, the security level for any given memory block within
19 the nonvolatile user memory 216 can be modified by action of processor 206 under
20 program control or under control of tester/programmer 220. One embodiment of
21 a process used to determine whether or not a transition in security level is
22 permitted is illustrated as process 500 of **FIGURE 5** which starts at 504. When an
23 instruction is received at 512, processor 206 determines if the instruction is for a
24 change in security level at 518. If not, normal operation proceeds at 522 and
25 control returns to 512 to await the next instruction (either from tester programmer
26 220 or by operation of another program on processor 206). If, at 518, the
27 instruction directs a security level change, the memory block to be changed is
28 identified at 526 and the current security level of the memory block is read at 532
29 from the supervisor nonvolatile memory 218. In the current exemplary embodiment,

1 transitions in security level are only permitted if a transition increases security.
2 Therefore, at 538 the current security level and the proposed new security level are
3 examined to determine if the new security level is higher than the old level. If so,
4 the security level is modified at 544 and control returns to 522. If, however, the new
5 security level is not higher than the old security level at 538, the change is rejected
6 and an error condition is flagged at 550. Depending on the nature of the
7 operational program of the microcontroller 200, such error conditions can result in
8 any number of operational steps including disabling the microcontroller 200.

9 In the above example, it is presumed that a hierarchy of security levels is
10 established so that it is clear that there is a higher level of security for one security
11 level than for another in all cases. However, the above process can be generalized
12 by process 600 of **FIGURE 6**. When an instruction is received at 512, processor
13 206 determines if the instruction is for a change in security level at 518. If not,
14 normal operation proceeds at 522 and control returns to 512 to await the next
15 instruction (either from tester programmer 220 or by operation of another program
16 on processor 206). If, at 518, the instruction directs a security level change, the
17 memory block to be changed is identified at 526 and the current security level of
18 the memory block is read at 532 from the supervisor nonvolatile memory 218. At
19 610, it is determined if the transition is permitted under the rules and algorithm
20 defined for permissible transitions of security level. If the transition in security level
21 is permitted at 616, then the security level is modified at 544 and control returns to
22 522. If, however, the transition is not permitted at 616, the change is rejected and
23 an error condition is flagged at 550. Depending on the nature of the operational
24 program of the microcontroller 200, such error conditions can result in any number
25 of operational steps including disabling the microcontroller 200.

26 In accordance with the above description, the present invention is used to
27 protect memory locations in a nonvolatile user program memory 216. The rules are
28 stored in the nonvolatile supervisory memory 218 and the algorithm for applying the
29 rules is stored in the supervisor ROM 244. However, this arrangement should not
30 be considered limiting since this invention can be configured in other ways without

1 departing from the invention. For example, in one embodiment both the rules and
2 the algorithm could be stored in ROM. In other embodiments, the protected
3 memory could be random access memory or read only memory or nonvolatile
4 Flash memory or a combination thereof. Such embodiments are considered
5 equivalents for purposes of this invention.

6 Those skilled in the art will recognize that the present invention has been
7 described in terms of exemplary embodiments based upon programming
8 nonvolatile memory within a microcontroller; however, the present invention should
9 not be so limited. The present invention could be implemented using hardware
10 component equivalents such as special purpose processors, micro-processors and
11 the like which are equivalents to the invention as described and claimed.

12 The present invention is preferably implemented using a programmed
13 processor executing programming instructions that are broadly described above in
14 flow chart form. Such instructions can be stored in any suitable electronic
15 programming medium. However, those skilled in the art will appreciate that the
16 processes described above can be implemented in any number of variations and
17 in many suitable programming languages without departing from the present
18 invention. For example, the order of certain operations carried out can often be
19 varied, and additional operations can be added without departing from the
20 invention. Error trapping can be added and/or enhanced and variations can be
21 made in user interface and information presentation without departing from the
22 present invention. Such variations are contemplated and considered equivalent.

23 The present invention provides enhanced security by virtue of isolating not
24 only the control lines, data lines and address lines of the nonvolatile memory from
25 the tester/programmer but also isolates the tester/programmer from the actual
26 process used to effect the programming or re-programming of the nonvolatile
27 memory 216. Moreover, in order to enter a programming mode, the security
28 procedures described above are used so that only modifications to the content of

1 the nonvolatile memory that are within prescribed security limits can be carried out.
2 Thus, the invention as described provides substantially enhanced security against
3 intruders attempting to program or re-program the nonvolatile memory 216.

4 Those skilled in the art will recognize that the present invention has been
5 described in terms of exemplary embodiments based upon programming
6 nonvolatile memory within a microcontroller; however, the present invention should
7 not be so limited. The present invention could be implemented using hardware
8 component equivalents such as special purpose processors, micro-processors and
9 the like which are equivalents to the invention as described and claimed.
10 Moreover, although described in connection with programming a nonvolatile
11 memory such as a Flash memory, the technique could equally well be used to
12 program a region of volatile memory such as RAM memory without departing from
13 the present invention. Also, although a two wire bus interface with the
14 tester/programmer is preferred, this is not to be limiting. Although the exemplary
15 embodiment of microcontroller 200 shows memories 216 and 218 sharing the
16 same data, address and control paths, this too should not be considered limiting
17 since other physical arrangements are also possible.

18 The present invention is preferably implemented using a programmed
19 processor executing programming instructions that are broadly described above in
20 flow chart form. Such instructions may be stored in any suitable electronic
21 programming medium. However, those skilled in the art will appreciate that the
22 processes described above may be implemented in any number of variations and
23 in many suitable programming languages without departing from the present
24 invention. For example, the order of certain operations carried out can often be
25 varied, and additional operations may be added without departing from the
26 invention. Error trapping may be added and/or enhanced and variations may be
27 made in user interface and information presentation without departing from the
28 present invention. Such variations are contemplated and considered equivalent.

29 While the invention has been described in conjunction with specific
30 embodiments, it is evident that many alternatives, modifications, permutations and

variations will become apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended that the present invention embrace all such alternatives, modifications and variations as fall within the scope of the appended claims.

What is claimed is:

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---